

FIG. 1

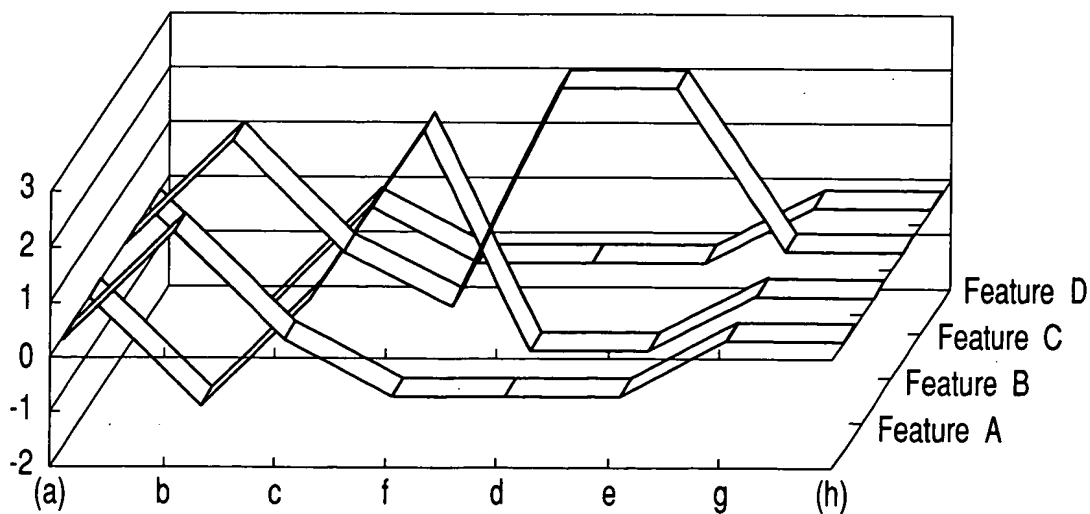


FIG. 3

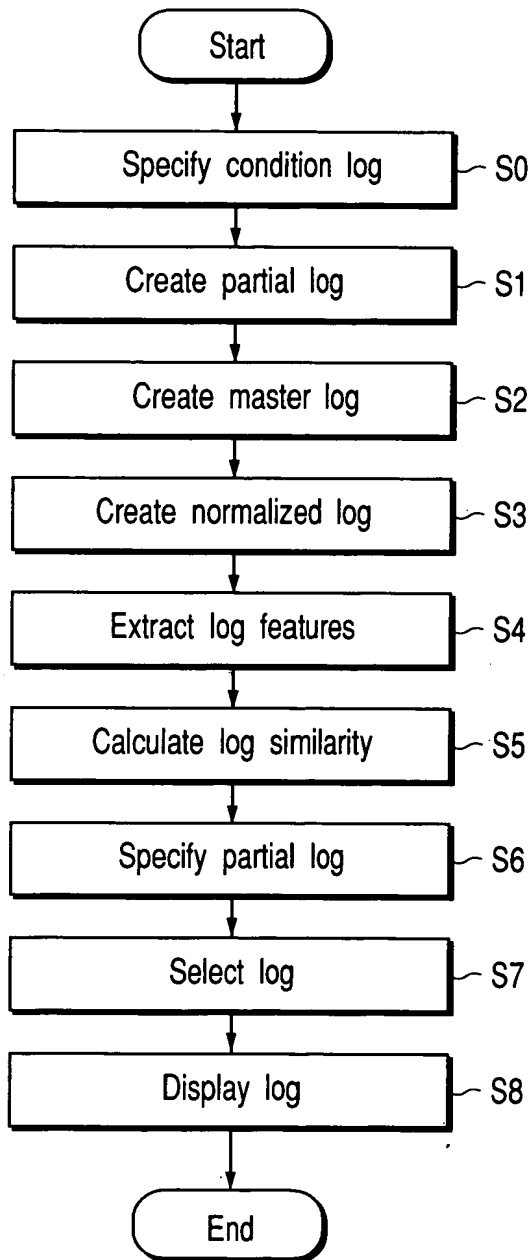


FIG. 2

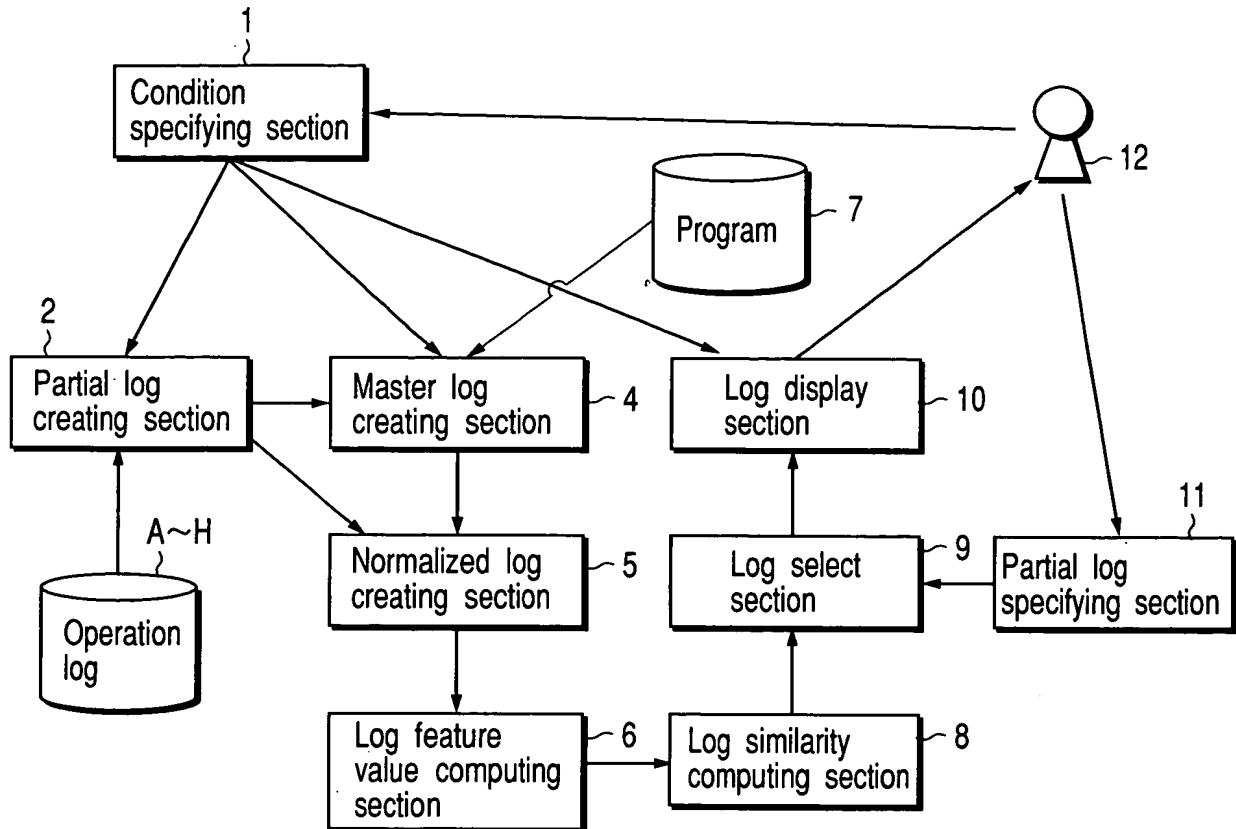


FIG. 4

```
#include<stdio.h>
#define MAX_STR_NUM 10

// declaration of functions
void PrintError();
void reverseString(char* string);
void makeValueString(int value,char* str);
// main functions
int main(int argc,char** argv){
    char string[MAX_STR_NUM];
    // convert the number of arguments into character strings
    // in ternary representation (the order of characters is in reverse)
    makeValueString(argc,string);
    // reverse the order of characters
    reverseString(string);
    // display the result
    printf(" %d to %s\n",argc,string);
}

// definition of each function
void makeValueString(int value,char* str);
{
    // recursion end condition for recursive function
    if(value<=0)
    {
        str[0]='\0';
        return;
    }
    makeValueString( value/3,str+1);
    switch(value%3)
    {
        case 0:
            str[0]='\0'; // mistaken for '0'
            break;
        case 1:
            str[0]='1';
            break;
        case 2:
            str[0]='2';
            break;
        default:
            break;
    }
}
```

FIG. 5A

```
    }  
  }  
  void PrintError()  
  {  
    printf("error\n");  
  }  
  void reverseString(char* string)  
  {  
    char tmp_char  
    int n;  
    int i;  
    n=strlen(string);  
    if(n==0)  
    {  
      PrintError(); // error process  
    }  
    else  
    {  
      // reverse the order of the character strings  
      for (i=0;i<(n/2);i++)  
      {  
        tmp_char=string[i];  
        string[i]=string[n-1-i];  
        string[n-1-i]=tmp_char;  
      }  
    }  
  }  
}
```

FIG. 5B

```
main(12,0x10000)
{
    makeValueString(12,0x20000)
    {
        if(value<=0)
        makeValueString(4,0x20001)
        {
            if(value<=0)
            makeValueString(1,0x20002)
            {
                if(value<=0)
                makeValueString(0,0x20003)
                {
                    if(value<=0)
                    {
                    }
                    }
                switch(value%3)
                {
                    case 1:
                    {
                    }
                    }
                switch(value%3)
                {
                    case 1:
                    {
                    }
                    }
                switch(value%3)
                {
                    case 0:
                    {
                    }
                    }
            }
        reverseString(0x20000)
        {
            n=strlen(0x20000);
            if(n==0)
            {
                PrintError()
                {
                    printf(" error\n");
                }
            }
        }
        printf(" %d to %s\n",12,0x20000);
    }
}
```

FIG. 6

```
main(13,0x10000)
{
    makeValueString(13,0x20000)
    {
        if(value<=0)
        makeValueString(4,0x20001)
        {
            if(value<=0)
            makeValueString(1,0x20002)
            {
                if(value<=0)
                makeValueString(0,0x20003)
                {
                    if(value<=0)
                    {
                        switch(value%3)
                        {
                            case 1:
                            {
                                switch(value%3)
                                {
                                    case 1:
                                    {
                                        switch(value%3)
                                        {
                                            case 1:
                                            {
                                                reverseString(0x20000)
                                                {
                                                    n=strlen(0x20000);
                                                    if(n==0)
                                                    else
                                                    {
                                                        for(i=0,i<(n/2);i++)
                                                        {
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    printf(" %d to %s\n",13,0x20000);
}
```

FIG. 7

```
int main(int argc, char** argv)
{
    makeValueString(int value, char* str)
    {
        if(value <= 0)
        {
        }
        makeValueString(int value, char* str)
        {
            if(value <= 0)
            {
            }
            makeValueString(int value, char* str)
            {
                if(value <= 0)
                {
                }
                makeValueString(int value, char* str)
                {
                    if(value <= 0)
                    {
                    }
                    makeValueString(value/3, str+1);
                    switch(value%3)
                    {
                        case 0:
                        case 1:
                        case 2:
                    }
                }
            }
        }
        switch(value%3)
        {
            case 0:
            case 1:
            case 2:
        }
    }
    switch(value%3)
    {
        case 0:
        case 1:
        case 2:
    }
}
```

FIG. 8A


```
switch(value%3)
{
    case 0:
    case 1:
    case 2:
    }
}
reverseString(char* string)
{
    n=strlen(string);
    if(n==0)
    {
        PrintError()
        {
            printf(" error\n");
        }
    }
    else
    {
        for(i=0,i<(n/2);i++)
        {
            for(i=0,i<(n/2);i++)
            {
                for(i=0,i<(n/2);i++)
                {
                }
            }
        }
    }
    printf(" %d to %s\n",argc,string);
}
```

FIG. 8B

FIG. 9A

```
1      }
1      switch(value%3)
1      {
1      case 0:
0      case 1:
0      case 2:
1      }
1  }
1  reverseString()
1  {
1      n=strlen();
1      if(n==0)
1      {
1          PrintError()
1          {
1              printf();
1          }
1      }
0      else
0      {
0          for(i=0,i<(n/2);i++)
0          {
0              for(i=0,i<(n/2);i++)
0              {
0                  for(i=0,i<(n/2);i++)
0                  {
0                      }
0                  }
0              }
0          }
0      }
1      printf();
1  }
```

FIG. 9B

[illegible]

[illegible]

	A	B	C	D	E	F	G	H
A		494	-130	-346	-90	-138	-26	-306
B			-178	-330	-138	-58	-74	-290
C				-58	70	22	70	-18
D					-82	-66	-146	470
E						126	110	-170
F							62	-154
G								-170
H								

FIG. 12